



مجلة شهرية تهتم بنظام لينكس والمصادر المفتوحة
العدد الخامس-مارس٢٠٠٨

في هذا العدد

ترجم مع STARDICT

تعرف على Ftp & OpenSSH

البرمجة بلغة python

تلميحات وافكار تفيدك

اخبار متنوعة وجديدة

تركيب البرامج في التوزيعات الديبانية

* والمزيد داخل هذا العدد

بسم الله الرحمن الرحيم



مجلة
لينكس العرب

مجلة شهرية شاملة تهتم بأمور الينكس خاصة والمصادر المفتوحة بشكل عام.

العدد الخامس مارس ٢٠٠٨

داخل هذا العدد





اعد موضوعات هذا العدد
أعضاء فريق لينكس العرب

موقع لينكس العرب
www.linux-ar.com



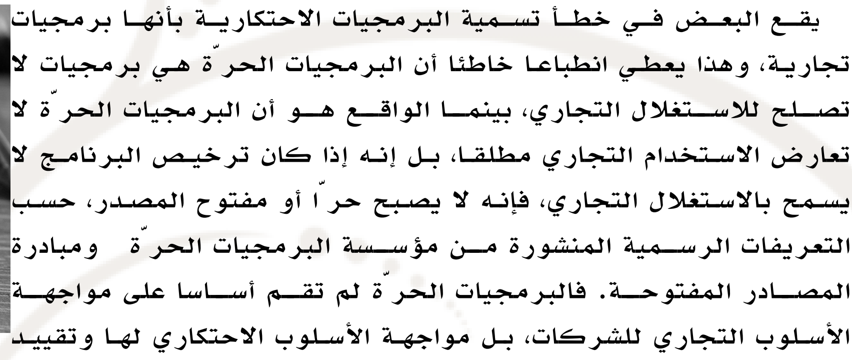
رئيس التحرير
حسن إبراهيم ضوّة

الاخراج والتحرير
أحمد السويلم
عبدالله الدعجاني

للمساهمة في الاعداد القادمة
http://linux-ar.org/forum
او على العنوان التالي
linux1010@gmail.com

التجارة والاحتكار لايستويان

إعداد: حسن إبراهيم ضوّة



يقع البعض في خطأ تسمية البرمجيات الاحتكارية بأنها برمجيات تجارية، وهذا يعطي انطباعا خاطئا أن البرمجيات الحرّة هي برمجيات لا تصلح للاستغلال التجاري، بينما الواقع هو أن البرمجيات الحرّة لا تعارض الاستخدام التجاري مطلقا، بل إنه إذا كان ترخيص البرنامج لا يسمح بالاستغلال التجاري، فإنه لا يصبح حرّا أو مفتوح المصدر، حسب التعريفات الرسمية المنشورة من مؤسسة البرمجيات الحرّة ومبادرة المصادر المفتوحة. فالبرمجيات الحرّة لم تقم أساسا على مواجهة الأسلوب التجاري للشركات، بل مواجهة الأسلوب الاحتكاري لها وتقييد حريات المستخدمين. فعند مراجعة تعريف البرمجيات الحرّة نجد أن استخدام البرمجيات الحرّة في تقديم خدمات تجارية أو بيع البرمجيات لا يعارض مبادئ البرمجيات الحرّة مادام من سيستقبل تلك البرمجيات سيحصل على حرياته الأساسية كما حددتها حركة البرمجيات الحرّة. فما يميز مجتمع تطوير البرمجيات الحرّة هو القدرة على المشاركة بأهداف مختلفة، فالمؤسسات العلمية والشركات التجارية وهواة البرمجة والتطوير يتمكنون من المشاركة في مجتمع واحد مع اختلاف أهدافهم واستفادتهم من البرمجيات الحرّة. وتقدم البرمجيات الحرّة مجتمعا تظهر فيه روح التنافس بدلا عن العداوة، ففي مجتمع لا يحتوي إلا البرمجيات الحرّة، تتنافس الشركات في تقديم خدمات ومنتجات بأسعار وعروض تنافس بها شركات أخرى في نفس المجال، فالأسلوب الاحتكاري يعتمد بشكل كبير على قوانين الملكية الفكرية وحماية الشركة من انتهاكات الشركات أو المستخدمين لتلك القوانين، الأمر الذي يهدر الكثير من الوقت والموارد لدى الشركات، ويخلق جوّ عدائي في أسواق الشركات الاحتكارية والغير احتكارية أيضا. ولذلك فإن الشركات بدأت تدرك أن انفتاحها وتعاونها مع المجتمع وتقدير حرية اختياره هي أساس النجاح للنشاطات التجارية، وأن الفكر الاحتكاري سيزول بفعل الزمن. فمع وجود الشركات التي تقوم تجارتها على البرمجيات الحرّة يزداد إدراك عملاء المنتجات الاحتكارية الموجودة بالفعل أن ما يحصلون عليه غير كاف، ولا يقدم لهم ضمانات وحقوق تسمح للعملاء بضمان استقلال أعمالهم وعدم الاعتماد على منتجات احتكارية تضعهم تحت طائلة رخصة مقيّدة بشروط لا يجب أن يقبلوا بها.

الشركات والمؤسسات التجارية هي عامل هام في نهضة البرمجيات الحرّة وتطويرها، ولا يجب نبذ أهميتها بأي حال من الأحوال، ليس لقوة نفوذ تلك الشركات، ولكن لأن أغراضهم وإن كانت تجارية بحتة فإنها ليست بالضرورة معارضة لمبادئ حرية البرمجيات، وطالما أنهم يقدمون تجارة لا تعارض تلك المبادئ، فإن هذا من شأنه المساهمة في نشر البرمجيات الحرّة ودعم تطويرها. فمجتمع البرمجيات الحرّة ليس مجتمعا قائما على العمل الخيري والتطوعي، بل هو مجتمع قائم على مبادئ الحرية والمشاركة، وليس في هذا ما يتناقض مع الأغراض التجارية بأي حال من الأحوال.

****

يكتبها: حسن ضوّة



انتهاء فعاليات مؤتمر فوسديم ٢٠٠٨

انتهت فعاليات مؤتمر البرمجيات الحرية الأوروبي لعام ٢٠٠٨ في ٢٤ فبراير ٢٠٠٨ وأعلنت فيه شركة أيه.إم.دي عن خطة مستقبلية لإصدار مواصفات ومكتبات برخصة مفتوحة المصدر بالتعاون مع مطورين من شركة نوفيل. وتم كشف الكثير من التحديثات المتعلقة بخادم النوفذة إكس.. ومشروعات معتمدة عليه. وتم الإعلان أيضا عن موعد إطلاق أول بطاقة رسومية مفتوحة المصدر بالكامل وسيتم نشر مواصفاتها وستكون متاحة للجميع برخصة جنو العمومية الشهر القادم.

صن تستحوذ على إنوتك

أعلنت صن عن صفقة شراءها لشركة إنوتك المنتجة لبرنامج البيئات الافتراضية Virtual Box المفتوح المصدر، وسيحظى البرنامج بناء على تلك الصفقة بدعم متزايد على الأجهزة التي تنتجها شركة صن لتوسيع نطاق الخدمات التي تقدمها الشركة لعملائها.

موقع البرنامج: http://virtualbox.org



الإعلان عن خطة تطوير Alien Arena 2008

تم الكشف عن أحدث إصدارات لعبة Alien Arena بتحديثات عديدة لكل النماذج الرسومية وأداء أعلى، ولأول مرة ستعمل اللعبة على أنظمة ماك أو.إس، كما سيشهد هذا الإصدار معاملات لعب أكثر توازنا والكثير من التعديلات التي تم اقتراحها من الكثير من اللاعبين.

دراسة عن البرمجيات الحرة بتكلفة مليون دولار

سيقوم الاتحاد الأوروبي بتمويل دراسة بمليون دولار أمريكي لدراسة البرمجيات مفتوحة المصدر واستخدامها في أفريقيا ، أسيا وأمريكا اللاتينية. وتهدف الدراسة إلى بحث واستكشاف احتياجات البرمجيات الحرة المفتوحة المصدر كأدوات تطوير في تلك البقاع. تشارك عدة مؤسسات وشركات مثل كانونيكال مقدمة الدعم الفني لتوزيعة أوبنتو، بالإضافة لعدة جامعات في هذه الدراسة التي من المفترض أن تستغرق سنتين لإعداد خريطة طريق واضحة لتطوير وتوسيع استخدامات البرمجيات الحرة.

الموقع الخاص بالمشروع: flossinclude.eu

جوجل تبدأ استقبال طلبات دعم المشروعات في مارس



أعلنت جوجل أنها ستبدأ تلقي طلبات المشروعات الـتي سيتم تطويـرها في Summer Of Code 2008 بداية من الرابع من مارس وحتى ٢٤ مارس. وهذه هي السنة الرابعة لهذا المشروع الذي تدعمه جوجل لتطوير البرمجيات الحرة.

الموقع: code.google.com/soc/2008

الإعلان عن بدء تنظيم يوم حرية المستندات

Document Freedom Day '08 March 26th

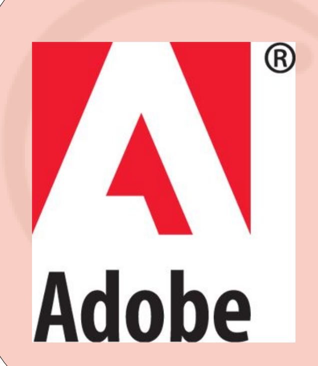
تـم تحديد يوم ٢٦ مارس ليكون يوم حرية المستندات حيث يتم عقد احتفاليات وندوات لدعم صيغة الملفات المفتوحة OpenDocument والمسجلة كمعايير عالمية بمنظمة الآيزو. أعلنت عدة شركات دعمها للحدث مثل جوجل وصن وآي.بي.إم وريدهات، بالإضافة لمؤسسة البرمجيات الحرة بفرعيها إضافة إلى الفرع الرئيسي.

الموقع: documentfreedom.org

شركة أدوب تطلق موقعها الخاص بالتقنيات المفتوحة المصدر

أطلقت شركة أدوبي موقعا خاصا لتتبع نشاطات الشركة بالمجتمع المفتوح المصدر، ومتابعة لأخبار آخر إصدارات الشركة ومساهماتها في البرمجيات المفتوحة المصدر. تم افتتاح الموقع في ٢٤ فبراير ٢٠٠٨ لكنه تضمن عدة إعلانات عن مساهمات الشركة بمشاريع dojo وsqlite.

الموقع: opensource.adobe.com







ترجم مع STARDICT

يكتبه :إبن يس

برنامج stardict هو برنامج ترجمة تتلخص مميزاته .

1 – مجانى و مفتوح المصدر .

2 – يعمل على ويندوز و لينكس .

3 – بسيط و خفيف و آمن.

4- يتوفر له العديد من القواميس و إمكانية التحويل من أى قاموس له .

5- إمكانية البحث عن الكلمات و إمكانية الترجمة بالتعليم على الكلمة فقط .

6- إمكانية الترجمة من مترجمات الإنترنت مباشرة

التركيب

البرنامج متوفر فى مصادر أغلب التوزيعات كل ما عليك هو البحث عن stardict فى مدير الحزم لديك أو تنزيله من الموقع و تركيبه من المصدر .

http://stardict.sourceforge.net

بعد تركيبه لا يحوى أى قواميس فتعال نعرف كيف نضيف له قواميس عربية .

أولا تحتاج إلى تنزيل قواميس عربية وتوجد بموقع babylon الشهير المختص بالترجمة

http://www.babylon.com/category/3/Arabic.html

قم بتنزيل القواميس التى تحتاجها و تصفح الموقع ففيه العديد من القواميس للغات كثيرة

و لنفترض أنك ستقوم بتنزيل أول قاموسين و هما Babylon Arabic-English و Babylon English-Arabic

بعد تنزيلهم على سطح المكتب مثلا ضع الأول فى مجلد و قم بتسميته مثلا إلى ara2en و الثانى فى مجلد آخر و سمه en2ara ، نحتاج بعد ذلك إلى برنامج dictconv الذى سنحول به صيغ قواميس babylon إلى صيغ القواميس العادية dict، قم بتنزيله من هنا

http://ibnyassen.googlepages.com/dictconv-0.2.tar.bz2

بعد ذلك قم بفك ضغط الملف و أدخل إلى مجلد البرنامج من سطر الأوامر و نركبه من المصدر، و نحتاج قبل التركيب من المصدر إلى مكتبة libxml2-dev ستجدها فى مدير الحزم لديك و من سطر الأوامر بعد ذلك و أنت مستخدم جذر

```
./configure

make all install
```

و إستخدامه كالتالى من سطر الأوامر

```
dictconv test.BGL -o test.ifo
```

قم بالدخول إلى المجلد الذى به القاموس المراد تحويله و الذى سميته ara2en و قم بتسمية ملف القاموس إلى إسم سهل مثلا ara2en.BGL و من سطر الأوامر من داخل المجلد

```
dictconv ara2en.BGL -o ara2en.ifo
```

سينتج من هذا الأمر ثلاث ملفات ara2en.dict و ara2en.idx و ara2en.ifo

نحتاج إلى أداة dictzip ستجدها فى مدير الحزم لديك و من نفس المجلد فى سطر الأوامر نفذ الأمر التالى

```
dictzip ara2en.dict
```

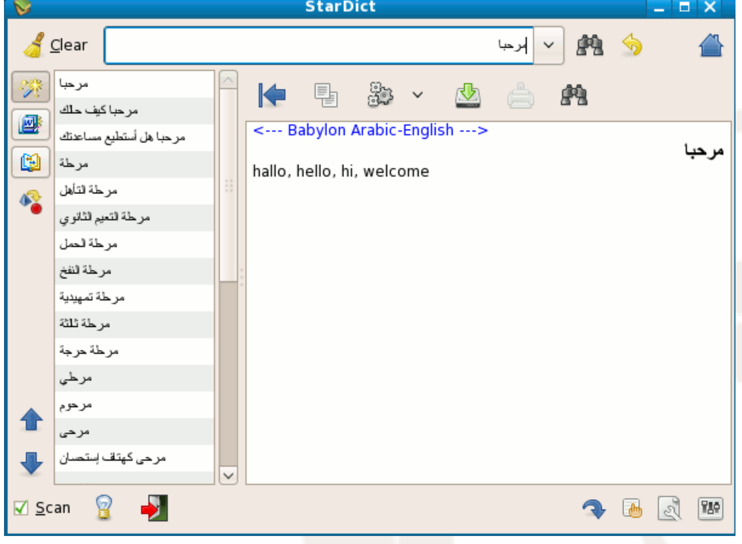
سيتم تحول الملف إلى ara2en.dict.dz و هذه الصيغة يقرأها قاموس stardict الآن نريد نقل المجلد بكل محتوياته إلى مكان تركيب القواميس للبرنامج و يمكنك حذف ملف ara2en.BGL أو الإحتفاظ به ..

و أنت فى نفس مكان التحويلات فى سطر الأوامر قم بالتراجع إلى المجلد السابق الذى يحوى المجلد المراد نقله بالأمر

```
cd .. # للتراجع إلى المجلد السابق
cp -r ara2en /usr/share/stardict/dic
```

مسار القواميس الذى يوضع به المجلدات الخاصة بالقواميس usr/share/stardict/dic/

قم بغلق البرنامج و فتحه مرة آخرى



علامة المنزل فيها قائمة التحكم بالقواميس و غيرها

قم بالتعليم على الكلمات لتظهر الترجمة كما فى الشكل التالي:-



كلمة scan أو مسح الموجودة فى أسفل على اليسار عند التعليم عليها يقوم بترجمة الكلمات بمجرد التظليل و يمكنك منع الخاصية .

فى بعض الأحيان تظهر رمز زائدة مع الترجمة و هذا يكون راجع لبرنامج التحويل dictconv لو واجهتك مشكلة فأطرحها فى منتدى لينكس العرب لنناقش حلها كما سنفتح موضوع لعمل التحويلات للقواميس و تجهيزها للإستخدام مباشرة و القواميس المعدة بهذه الطريقة تصلح لنظام التشغيل ويندوز مع البرنامج .

*****

OpenSSH & FTP

يكتبه :محمد عبدالله

من مهام مدير النظام اليومية الدخول الى عدد من الأجهزة والخوادم ، عمل ترقية لها أو حل مشكلة ما أو حتى صيانتها . تخيل معي هذا الشخص المسكين وهو ينتقل من غرفة الى اخرى أو من طابق الى آخر !!! كيف سيكون الوضع ؟! ألا توجد طريقة أسهل والأهم من ذلك طريقة آمنة للقيام بكل هذه المهام دون التحرك من على مكتبه ومن أمام شاشة جهازه ؟

هذا ما سنتطرق له بحول الله في هذا الموضوع كما سنتطرق لبروتوكول نقل الملفات الأقدم FTP . ونرجوا ملاحظة أن الشرح سيكون على نسخة RHEL 4 وما فوق .

أولاً : openSSH

SSH :

1 – نبذه :

قد تكون سمعت عن telnet ، انه شبيه بـ SSH والتي هي اختصار Secure Shell ولكن الفرق الكبير والجوهري بين الاثنين هو في البيانات عند نقلها عبر الشبكة . تكون البيانات عند نقلها عن طريق telnet غير مشفرة مما يجعلها صيد ثمين وسهل لأي مستخدم في الشبكة أو الانترنت ، وبالإمكان الحصول على بيانات مثل كلمات المرور بكل سهوله . أما SSH فتكون البيانات مشفرة .

SSH هو بروتوكول يسمح بتبادل الملفات بين جهازين عبر قناة مشفرة . يستخدم الـ SSH المفتاح العام Public-Key للتحقق من المستخدمين .

يستخدم SSH عادة للدخول الى جهاز بعيد وتنفيذ عدد من الأوامر عليه ويمكنه أيضاً استخدام X11 لتشغيل الواجهه الرسومية على جهاز بعيد .كما يستخدم أيضاً لنقل الملفات باستخدام SCP أو SFTP .

يستخدم خادم SSH المنفذ 22 من خلال البروتوكول TCP . ويمكن للـ Clients استخدام الأمر ssh للاتصال بخادم SSH . يتوفر SSH على عدد كبير من الأجهزة وأنظمة التشغيل أمثال FreeBSD ,Linux,Solaris,Mac OS .

تم تصميم SSH عام 1995 على يد الباحث Tatu Ylönen من جامعة هلسنكي الفنلندية . وقد كان الهدف منه هو استبدال البروتوكولات القديمة telnet,rlogin,rsh ببروتوكول جديد يدعم تشفير الملفات .

2 – اعداد SSH .

الخدمة المسؤولة عن تشغيل SSH Server هي SSHD .

سنحتاج الى الثلاث البرامج التالية :

1. openssh وهو المسؤول عن تقديم برتوكول SSH .
2. openssh-server ويحتوي على الملفات اللازمة لتشغيل خدمة SSH .
3. openssh-clients ويحتوي على البرامج اللازمة للوصول الى SSH وبرامج اخرى مثل scp,sftp .

ملف اعداد الخدمة هو

```
/etc/ssh/sshd_config
```

ملف اعداد SSH يكون جاهزاً في أكثر التوزيعات ولكننا هنا

نريد أن نزيد من فعالية الخدمة وزيادة الأمان بها فمثلاً SSH تعمل على اصدارين من البروتوكولات 1 و 2 ، الإصدار الثاني أكثر أماناً وعلينا استخدامه ، قد نرغب مثلاً في تغيير المنفذ الذي تعمل عليه الخدمة من 22 الى أي 1500 وهكذا .

مثال على ملف اعداد SSH :

#Port 22

المنفذ الذي تعمل عليه الخدمة.

#Protocol 2

البروتوكول الثاني وهو الأفضل.

#ListenAddress 192.168.1.1

رقم الآي بي الذي تعمل عليه الخدمة وهو مهم خاصة في حال وجود أكثر من بطاقة شبكة.

Authentication:

#LoginGraceTime 2m

هذا الخيار يحدد دقيقتين ليتم بعدها قطع الإتصال الخامل.

#PermitRootLogin no

منع الرووت من الدخول الى الجهاز بصفة مباشرة.

#MaxAuthTries 2

تحديد عدد محاولتين لتسجيل كلمة المرور.

#X11Forwarding yes

السماح بتمرير برامج رسومية لعرضها على جهاز بعيد.

هذه أبرز ملامح اعداد ملف SSH .

3 – الإتصال بخادم SSH .

للإتصال بخادم SSH استخدم الصيغة التالية :

```
ssh username@hotname-or-ip
```

حيث :

username هو اسم المستخدم الذي تريد استخدامه للدخول للجهاز البعيد وفي حال عدم كتابة اسم مستخدم فإن النظام سيفترض بأن اسم المستخدم هو نفس المستخدم على جهازك الحالي .

Hostname-or-ip هو اسم الجهاز الذي تريد الوصول له وفي حال عدم وجود DNS في الشبكة التي انت عضو بها استخدم رقم IP الخاص بالجهاز الخادم .

ملاحظة :

لا تستخدم أبداً المستخدم root للوصول الى جهاز بعيد ، استخدم اسم مستخدم عادي وبعد الدخول الى الجهاز الخادم قم برفع صلاحيتك الى المستخدم root ، ولهذا قمنا في ملف الاعداد بمنع المستخدم root من الوصول الى هذه الخدمه بصفه مباشرة .

في حال كنت قد غيرت المنفذ من 22 الى أي رقم آخر ، يمكنك تحديد المنفذ الذي تريد الاتصال به عن طريق الخيار p .

عندما يكون الخادم يسمح بتمرير برنامج X11 فاستخدم الخيار X عند الاتصال بالخادم وذلك لكي تستطيع تشغيل برامج رسومية الى الخادم ، انظر المثال التالي :

```
ssh -X -p 1500 mohammed@192.168.1.1
```

تشغيل الخدمة :

الخدمة المسؤولة عن تشغيل SSH هي SSHD ، ولتشغيل الخدمة نفذ الأمر التالي :

```
service sshd start
```

أو

```
/etc/init.d/sshd start
```

ولجعل الخدمة تعمل عند بدء التشغيل بصورة تلقائية نفذ

```
chkconfig sshd on
```

: SCP

وهي اختصار Secure CoPy أي النسخ الآمن ، عن طريق هذا الأمر يمكنك نسخ ملفاتك بأمان من جهاز الى جهاز آخر ، وصيغه الأمر كالتالي :

```
scp localfile username@hostname:/var/ftp/pub/upload/
```

حيث :

localfile هو الملف الذي تريد ارساله الى الخادم

/var/ftp/pub/upload/ هو المسار الذي نريد ارسال الملف له ، وفي حال كنت تريد ارسال ملف الى مجلد الهوم الخاص بك على الخادم فإنك لست بحاجه الى تحديد username كما ذكرنا سابقاً عندما تحدثنا عن ssh ، أيضاً لست بحاجه لكتابة المسار بعد : فيصبح الأمر كالتالي :

```
scp localfile hostname:
```

يمكنك أيضاً نسخ ملف من على الخادم الى جهازك مباشرة وذلك عن طريق الصيغة التالية :

```
scp username@hostname:/var/ftp/pub/database.db .
```

سنقوم هنا بنسخ ملف قاعدة البيانات database.db من مجلد ftp على الخادم الى الدليل الحالي ، لاحظ أن " . " تعني الدليل الحالي . ونفس ماذكرناه عن الأمر السابق ينطبق هنا .

في حال أردت نقل مجلد من أو الى الخادم استخدم الخيار r ، شاهد المثال التالي :

```
scp -r phpBB/ username@hostname:/var/www/html/
```

: SFTP

وهو بروتوكول نقل للملفات يعمل نفس عمل ftp ولكنه يقوم بنقل البيانات بطريقة مشفرة . للإتصال بجهاز عن طريق sftp استخدم الصيغة التالية :

```
sftp username@hostname
```

لعرض قائمة بالخيارات المتاحة اكتب الأمر help بعد الدخول الى الجهاز البعيد ، ومن هذه الخيارات :

الخيار	الغرض
lls	لعرض الملفات الموجودة على الجهاز المحلي وليس البعيد
ls	لعرض الملفات الموجودة على المسار الحالي في الجهاز البعيد
put	لنقل ملف من الجهاز المحلي الى البعيد
mput	لنقل عدة ملفات من الجهاز المحلي الى البعيد
get	للحصول على ملف من الجهاز البعيد الى المحلي
mget	للحصول على عدة ملفات من الجهاز العيد الى المحلي
cd	للتنقل بين المجلدات

ونفس ماذكرناه بخصوص ssh و scp ينطبق على sftp .

ثانياً : FTP .

FTP وهـي اختصـار File Transfer Protocol أي بروتوكول نقـل الملفات ، وهـو مـن أقدم بروتوكولات نقـل الملفات الموجودة الآن . هناك أكثر من خادم FTP ولكننا هنا سنتحدث عن VSFTPD وهي اختصار Very Secure File Transfer Protocol Daemon .

الحزمة المسؤولة عن هذه الخدمة هي vsftpd ، قم بتثبيتها من أقراص توزيعتك أو عن طريق RHN إذا كنت مشتركاً مع Red Hat .

الملف المسؤول عن اعداد FTP هو :

```
/etc/vsftpd/vsftpd.conf
```

ملاحظة هامة جداً :

قد يكون ملف الإعداد مربكاً بعض الشيء ، فمن المعروف في عالم اللينكس واليونيكس أنه عند وضع علامة # في ملف الإعداد فإن ذلك يعتبر تعليق مما يعني أن الخدمة لا تقرأة أي أن الخيار الذي قمت بإلغاءه لن يعمل ،ولكن هنا يوجد اعدادات افتراضية تعمل بصورة تلقائية فمثلاً عندما تقوم بإلغاء الخيار anonymous_enable=YES أي تقوم بوضع # قبله فإنك بهذا تريد منع المستخدم مجهول الهوية من الولوج الى خدمة FTP لديك ، ولكن هنا وبعد عمل هذا ستجد أي مستخدم يدخل الى الخدمة بدون اسم مستخدم وكلمة مرور !!! لماذا ؟ لأن الوضع الإفتراضي هو السماح للمستخدمين كافة بالدخول الى الخادم ، إذاً ماذا تعمل ؟ عليك بجعل القيمة NO . حسناً وكيف تعرف الإعدادات الإفتراضية ؟ راجع manual الخاص بالخدمة عن طريق الأمر man vsftpd.conf وهذا مثال منه :

```
anonymous_enable
    Controls whether anonymous logins are permitted or
not If enabled, both  the  usernames  ftp  and
       anonymous are recognised as anonymous logins.


       Default: YES
```

ملف إعداد FTP سهل وواضح وقد يكون أهم أمر فيه هو تفعيل الخاصية chroot .

ماهي خاصية chroot ؟

عندما يقوم أحد المستخدمين المحليين لديك بالولوج الى حساب FTP فسيكون بإمكانه الولوج الى ملفات النظام وأخذ ما يريد منها وهذا خطر كبير جداً بل طامة كبرى . يقوم هذا الخيار على حجز المستخدمين في منازلهم وعدم مغادرتهم لها بأي حال من الأحوال .

لتفعيل هذه الخاصية عليك اظافة التالي الى ملف الإعداد (هذا الخيار غير موجود افتراضياً)

```
chroot_local_user=YES
```

الآن أعد تشغيل الخدمة .

هناك خيار آخر يمكنك من عمل استثناء لبعض المستخدمين وهو

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
```

أولاً قم بإنشاء الملف chroot_list في المسار الموضح كما في ملف الإعداد ، ثانياً تأكد من تفعيل خيار حجز المستخدمين المحليين ، ثم أظف المستخدمين الذين تريد استثناءهم من هذا الحجز الى الملف chroot_list .

والى لقاء قادم بمشيئة الله تعالى,,,.

***

لغة البرمجة بايثون Python

يكتبه – الشريف أحمد حسن خلف

قبل البدء

تكلمنا فى المقال السابق عن المتغيرات و الثوابت و أدوات الإدخال و العمليات الحسابية ، وفى هذا المقال سنتكلم عن الدوال و كيفية استخدامها و الاستفادة منها و أهميتها مع ضرب الأمثلة المتعددة فبالمثال يتضح المقال .

ترجمة كلمة Functions تعنى وظيفة أو دالة ؛ الدالة فى الرياضيات هى علاقة بين متغير تابع و متغير مستقل مثلا دالة تدرس العلاقة بين س المتغير المستقل و ص المتغير التابع .

```
ص = س - 1
# نكتبها كدالة هكذا
ص = د(س) = س - ١
```

يعنى عند تغيير قيمة س تتغير قيمة ص تبعا لهذا التغير فنقول ؛ د(5)= 5 - 1 إذا د(5) = 4 يعنى قيمة ص عندما نضع قيمة س بخمسة . الرقم الذى بين القوسين يسمى مدخل argument و النتيجة تسمى القيمة المرجوعة أو العائدة return value إذا الدالة تأخذ قيمة مدخلة و تعطيك نتيجة وهناك دالة تأخذ قيمتين مثلا :

```
دالة ج = د(س , ص ) = 6س + 7 ص
د(5 ,3) = 6 * 5 + 7 * 3
```

و قد تتعدد المدخلات و تكون دالة ج دالة فى (س و ص) و هذا تبسيط جدا لمفهوم الدوال فهى مجال واسع فى علم الرياضيات .

الدوال فى البايثون

لكل دالة إسم معين تستدعيها به فمثلا دالة تحديد النوع type

```
>>> type("32")
<type 'str'>
# لاحظ أنه حدد النوع كنص لانها بين علامتى تنصيص
>>> srings = type("32")
>>> print strings
<type 'str'>
>>> type(32)
<type 'int''>
```

كتبنا إسم الدالة و بعدها قوسين بينهما المدخل المراد تطبيق الوظيفة المعينة عليه و هى تحديد نوع المتغير .

```
>>> type(32.01)
<type 'float'>
```

هنا حدد المتغير 32.01 كرقم عشرى float .

دالة int تحول الرقم العشرى float أو الرقم الذى بين علامات تنصيص إلى رقم صحيح integer و لا تقبل النصوص.

```
>>> int("32")
32
>>> int(3.99999)
3
>>> int("World")
ValueError: invalid literal for int() with base 10: 'World'
```

وكذلك دالة float تأخذ متغير العدد الصحيح و تحوله إلى عشرى والعدد العشرى الذى بين علامتي التنصيص و لاتقبل النصوص أيضا .

```
>>> float(32)
32.0
>>> float("3.14159")
3.14159
```

ودالة str التى تغير نوع المتغيرات إلى نوع نصي.

```
>>> x=5
>>> x=str(x)
>>> type(x)
<type 'str'>
>>> str(5.555)
'5.555'
```

ما الفائدة من تحويل الرقم الصحيح إلى عشرى ؟
أكتب هكذا و تأكد من الناتج

```
>>> minute = 59
>>> minute/60
0
```

الناتج صفر لماذا ؟ لأن المفسر يقوم بإخراج النتائج بالأعداد الصحيحة فقط و ما الحل ؟ الحل هو جعل نوع المتغير minute كنوع عشرى بالدالة float .

```
>>> float(minute) / 60
0.98333333333333328
```

أو كتابة المقسوم عليه 60 هكذا 60.0 فيعرف المفسر أن نوع هذا المتغير رقم عشرى .

```
>>> minute = 59
>>> minute / 60.0
0.983333333333
```

هذه بعض دوال البايثون و هى كثيرة جدا .

الوحدة .... Module

الوحدة هـــى ملف يحتوى على مجموعـــة دوال لغرض معيـــن والوحدة يجـب عمل إسـتيراد لهـا ، مثلا يوجـد فـى البايثون وحدة الرياضيات بها مجموعة دوال للعمليات الرياضية ،مثل حساب الزوايا المثلثية " جا و جتا و ظا و .. " و كيفية إستيراد الوحدة بالأمر import و لابد من عمل إستيراد قبل تنفيذ أى دالة منها .

```
>>> import math
```

لإستخدام دوال الوحدة تكتب إسم الوحدة متبوعا بنقطة و بعدها إسم الدالة " و القوسين لإدخال المدخلات "

```
>>> x=math.log10(30.0)
>>> print x
1.47712125472
```

هذاالدالة لحساب لوغاريتم 30.0 للأساس 10 و لو لم نكتب الرقم 10 تصبح دالة اللوغاريتم للأساس الطبيعى e .

أفتح محرر النصوص المفضل لديك و أكتب هذا البرنامج البسيط:

```
import math
angle=input(" Enter your angle .." )
print "the result sin(your angle is " , math.sin(angle)
print "the result cos(your angle is " , math.cos(angle)
print "the result tan(your angle is " , math.tan(angle)
```

السطر الأول هو إستيراد للوحدة التى نريدها و السطر الثانى هو متغير إسمه angle بدالة الإدخال input و هى دالة إدخال لا تقبل إلا الأرقام فقط و السطر الثانى هو الأمر print و به دالة جيب الزاوية sin و هكذا إستخدمنا cos و tan . كما يمكن وضع مصطلح أو تعبير (expression) كمدخل(argument) للدوال و الدالة ما هى إلا تعبير فمثلا :

```
x = math.sqrt(math.log(10.0))
```

قمـــت بإدخال دالة آخرى داخـــل الدالة الأصـــلية فالدالة الأصـــلية math.sqrt () هى لحساب الجذر التربيعى للمدخل و المدخل عبارة عن دالة تحسب قيمة اللوغاريتم.

كيفية تعريف دالة جديدة

تســـتطيع بلغـــة البايثون أن تقوم بتعريـــف دالة "وظيفـــة " جديدة لإستخدامها و الفائدة من عمل دالة هى توفير للخطوات فتقوم بعمل دالة تنجز وظيفة معينة فيما بعد فهى تقلل أسطر البرنامج و تسهل عمل مراجعة و تطوير البرنامج و تتفوق بايثون فى هذا المجال بشكل ملحوظ فهـــى لغـــة مفضلة عنـــد علماء الرياضيات و الباحثيـــن و المهندسين و مصممى الرسـوم ثلاثيـة الأبعاد و برمجـة الألعاب فهـى أشهر اللغات فى برمجة الألعاب فهى قوية و سهلة .. تعال نتعرف عن كيفيـة تعريـف دالة جديدة داخـل البايثون . لماذا تسـمى بتعريـف ؟ لإنـك بكل بسـاطة تقوم بتعريـف المفسـر أن هذه دالة سيسـتخدمها و

تعريـف الدالـة يبدأ بكلمـة def ثـم إسـم الدالة ثـم قوسـين ثـم تنهيها بــ : و هذا يسمى رأس الدالة والجزء الثانى يبدأ فى السطر التالى للرأس بعـد أربعـة مسـافات مـن أول السـطر و تكون فيـه الأوامـر و البيانات المختلفـــة (STATEMENTS) فلنضرب مثال لتعريـــف دالة ..

```
def NAME( LIST OF PARAMETERS ):
   STATEMENTS
```

أسـماء الدوال تتبـع نفـس قوانيـن تسـمية المتغيرات التـى ذكرتهـا فـى المقال السـابق ، ولا تسـمى دالة بإسـم متغيـر قمـت بتسـميته أو دالة آخرى بنفس الاسم وبين القوسين يكون المدخل و فى السطر الثانى بعد أربع مسافات يبدأ جسم الدالة و يمكن و ضع عدد غير محدود من البيانات .

```
def your_name():
   print  " Your name"
```

هذه دالة تقوم بطباعـة سـلسـلة نصيـة Your name إسـم الدالة your_name و الأقواس الفارغة () تشير إلى أن هذه الدالة لا تأخذ مدخلات و السطر الثانى يحوى بيان (STATEMENT) للطباعة .

و كيفيـة استخدامها فـى ملف نصى (script) هكذا " قد سبق و بينت فـى المقال الأول الفرق بين كتابـة البرنامج فى ملف نصى script و إسـتخدام المفسـر التفاعلى للغـة البايثون بكتابـة كلمـة python فـى سطر الأوامر "

```
def your_name():
   print " Your name "
your_name()
```

وناتج هذا البرنامـج سيكون تنفيذ أمر الطباعة ، فاستخدام الدالة هو بكتابـة اسـمها متبوعا بقوسـين ، ولو كانـت تحتاج مدخلات فسـتكتب داخـل القوسـين كمـا سـنوضح واذ لم تكـن تحتاج فتكون كمـا هـى فارغة .

يمكـن إسـتدعاء دالة داخـل دالة آخرى ، وسـوف نرى فـى هذا المثال ولنفترض إن البرنامـج يحتوى على الدالة السـابقة أيضا ونضيـف دالة جديدة.

```
def my_name():
   print " My name is sherif "
```

و الدالة القادمـــة ســـتحوى الدالتيـــن الســـابقتين your_name و my_name .

```
def my_your_name():
   my_name()
   your_name()
```

ثم تقوم بتنفيذها فى سطر جديد my_your_name ()

سيكون شكل الملف النصى هكذا .

```
def your_name():
    print " Your name "
def my_name():
    print " My name is sherif "
def my_your_name():
    my_name()
    your_name()
my_your_name()
```

و الناتج سيكون

```
[sherif@localhost ~]$ python mathes.py
 My name is sherif
 Your name
```

لاحظ أنه عندما تستدعى دالة فمن المنطقى أن تكون معرفة مسبقا فالمفسر يبدأ البرنامج من السطر الأول ثم الثانى و هكذا فلابد من الترتيب فى تعريف الدوال و إستدعائها .

دوال بمدخلات argument

لتعريف دالة بمدخلات لابد بوضع بارمتر داخل القوسين و البارمترات هى كالمتغيرات لكنها قاصرة على الدالة فقط فمثلا.

```
def two_man(man):
    print man , man
```

لإستدعاء الدالة ووضع مدخلات مكان البارمتر man نكتب الدالة و بين القوسين أي قيمة بدلا من كلمة man و هذه الكلمة يمكن تغييرها لأى اسم.

```
two_man("sherif")
two_man(555)
```

الناتج بعدها.

```
sherif sherif
555 555
```

يمكن وضع بيان فى المدخل فمثلا.

```
two_man(6*4)
```

أو متغير سبق كتابته.

```
dist=" Fedora and Joatha "
two_man(dist)
```

سيكون الناتج طباعة المتغير مرتين .

و يمكن تعريف دالة بمدخلين مثلا نقوم بعمل دالة لجمع شيئين.

```
def cat_twice(part1, part2):
    cat = part1 + part2
    print cat
```

هنا سيأخذ قيميتن و لتنفيذ الدالة لابد من إدخال قيمتين عددية أو نصية.

```
cat_twice("me "," you")
cat_twice(5,10)
```

سيضم كلمتين me و you و سيجمع الرقمين 5 و 10 جرب أنت بنفسك ، وشكل الملف النصى سيكون هكذا.

```
def cat_twice(part1, part2):
    cat = part1 + part2
    print cat
cat_twice("me "," you")
cat_twice(5,10)
```

لاحظ هنا أن هناك متغير داخل الدالة إسمه cat حاول عمل أمر طباعة له print cat ستجد أن المفسر يعطيك خطأ معناه أن هذا المتغير لم يتم التعرف عليه .. لماذا ؟ لإن المتغيرات داخل الدوال محلية و قاصرة في داخل الدالة فقط و ليس لها أى علاقة بباقى البرنامج .

الدوال المنتجة و الغير منتجة

إستكمالا للبرنامج السابق قم بعمل متغير يساوي الدالة cat_twice(5,10) و سمه مثلا result .

```
result=cat_twice(5,10)
print result
```

ستجد أن قيمة result هى None لكن لماذا ؟ و لماذا الدوال الآخرى كدوال وحدة math تقوم بحفظ القيمة ؟

هناك نوعين من الدوال ، دالة منتجة ودالة غير منتجة فالدالة قاصرة على تنفيذها و لا تقوم بإرجاع قيمة معينة فى الذاكرة و لكن كى تقوم بإرجاع أو حفظ قيمة المخرجات تحتاج إلى وضع أمر return على return . سنكتب دالة تقوم بحساب مساحة الدائرة و قانون و سنرى بالمثال كيف نستخدمه و الدوال التى كتبنها لا تحتوى المساحة هو حاصل ضرب ط فى تربيع نصف القطر.

```
import math
def area(radius):
    temp = math.pi * radius**2
    return temp
area_c=area(5)
print area_c
```

فى هذا البرنامج لن يعطيك قيمة المتغير بـ None بل سيعطيك قيمة حقيقية ويمكن إختصار الدالة هكذا.

```
def area(radius):
    return math.pi * radius**2
```





ولكن السابقة أفضل عند مراجعة البرنامج وإكتشاف الأخطاء . و دالة math.pi هى قيمة باى أو ط أو 22/7 أو 3.14 .

إسطوانة

نريد كتابة برنامج يقوم بحساب حجم الإسطوانة .قانون حجم الإسطوانة هو عبارة عن حاصل ضرب مساحة القاعدة فى الإرتفاع و القاعدة هى دائرة . إذا المدخلات هى إرتفاع الإسطوانة و نصف قطر قاعدتها وأفضل طريقة لكتابة البرنامج هى التدرج مع البرنامج فلنبدأ بكتابة شكل الدالة . هى دالة ستحتاج إلى مدخلين الإرتفاع h و نصف القطر r و بما إن القاعدة دائرة فسنستخدم دالة حساب مساحة الدائرة و نبدأ بكتابة شكل الدالة .

```
import math
def area(radius):
    temp = math.pi * radius**2
    return temp
def cylender_valume(h,r):
    return 0.0
```

و قم بتجربة البرنامج أولا وتأكد من عدم وجود أخطاء و بعد ذلك سنبدأ بحساب مساحة القاعدة و هى بإستدعاء الدالة الأولى لحساب مساحة الدائرة و أعلم أنه أى سطر برمجى بعد جملة return ليس له أى قيمة فالإضافة ستكون قبلها فستصبح الدالة هكذا.

```
def cylender_valume(h,r):
    radius = area(r)
    return 0.0
```

و قم بتجربة البرنامج وتأكد من عدم وجود أخطاء و بعد ذلك نبدأ بحساب الحجم .

```
def cylender_valume(h,r):
    radius = area(r)
    valume = h * radius
    return 0.0
```

تأكدت أن البرنامج صحيح ؟ قم بعمل return للحجم و بعد ذلك نريد عمل مدخلات و تطبيق للدالة و إنهاء البرنامج.

و يصبح شكل البرنامج النهائى هكذا

```
import math
def area(radius):
   temp = math.pi * radius**2
   return temp
def cylender_valume(h,r):
   radius = area(r)
   valume = h * radius
   return valume

R = input(" Enter the radius of cylender " )
H = input(" Enter the height of cylender " )

valume = cylender_valume(R,H)
print valume
```

ولنا لقاء بمشيئة الله تعالى فى العدد القادم

***

دليل الحائرين في تركيب البرامج

" للتوزيعات الدبيانية "

يكتبه – المعتصم السرحاني

يعلم اغلبكم ان اكبر مشكله تؤرق المبتدئين وحتى متوسطي الاحتراف في نظام لينكس هو تركيب البرامج واصبحت الشغل الشاغل وبصراحة لم أجد موضوع واحد يتكلم بإسهاب في هذا الخصوص وإذا وجد نجده يتكلم عن معلومات قديمه غير مجدية بالوقت الحالي خصوصا مع التطور الكبير في التوزيعات وبالتحديد في مدير الحزم الذي اعتبره اعجوبة نظام التشغيل لينكس .

سيكون هذا الموضوع للتوزيعات المبنية على debian وبالخصوص على ubuntu .

طرق التركيب :

في نظام التشغيل لينكس وبالتحديد توزيعة ubuntu هناك ثلاثة طرق للتركيب .

1. مدير الحزم.
2. مثبت الحزم dpkg .
3. التركيب من المصدر .

1. مدير الحزم

تتميز التوزيعات الدبيانية (debian) بمديرحزم جبار وأعتقد انه الأقوى من باقي التوزيعات الأخرى مثل red hat و الأسهل بالتعامل مع وظيفة مدير الحزم في نظام لينكس بشكل عام ، ويقوم بتحميل البرامج من مصادر خارج النظام و تثبيتها سواء كانت هذه المصادر من CD or DVD او حتى الانترنت internet تجلب الحزم deb للتوزيعات debian وتقوم بعملية تثبيتها في النظام بمجرد وضع الأمر.

ولتشغيل مدير الحزم في ubuntu هناك طريقتان :

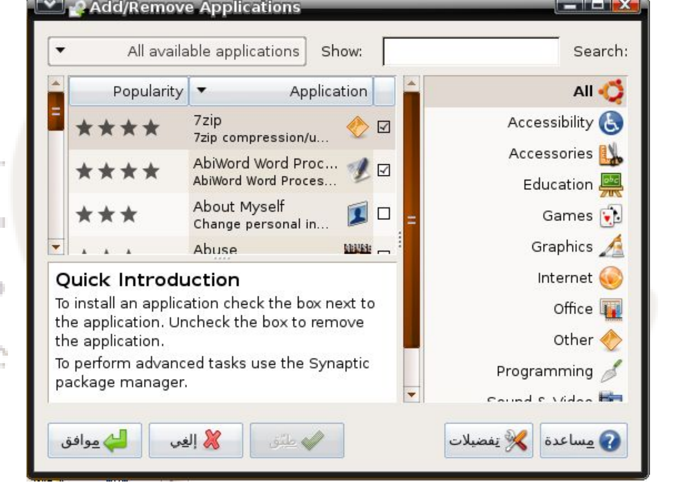
الأولى من خلال الواجهة الرسومية سواء كانت في KDE او GNOME واسم البرنامج Synapticومن خلال هذا البرنامج تستطيع التحكم بمصادر الحزم من تثبيت وإزالة فقط بأختيار البرنامج و البدء بالعملية ،وهناك برنامج اخر يقوم بالمهمه بطريقة أسهل و منظم بشكل أكبر وهو برنامج add/remove حيث ينظم البرامج فيه حسب تخصصها و الأقسام الموجودة فيه مثل الألعاب , البرمجة ,تعليمي , الفيديو و الصوت , انترنت و غيرها من الأقسام،والطريقة الأخرى في استخدام مدير الحزم هو من خلال سطر الأوامر ،سيتم شرحه لاحقا.

يقصد بمصادر الحزم هي الأماكن التي من خلالها يقوم مدير الحزم بالتحميل منها و تثبيتها في النظام مثل روابط لسيرفرات بالانترنت او cd او DVD .

صورة من synaptic



صورة من Add/Remove



والطريقة الثالثة لإدارة الحزم هو باستخدام سطرالأوامر

```
apt-get
```

حيث يتم استخدام برنامج apt-get لتركيب البرامج بهذا الأمر

```
sudo apt-get install "program"
```

حيث يوضع بدل "progarm" اسم البرنامج المراد تثبيته بالتوزيعة ولحذف البرنامج نستخدم الأمر

```
sudo apt-get remove program
```

و مصدر الحزم يكون في هذا الملف

```
sources.list
```

ولفتح هذا الملف استخدم الأمر التالي

```
sudo gedit /etc/apt/sources.list
```

من خلاله تستطيع التحكم بمصادر مديرالحزم

ولمستخدمي ubuntu وجدت هذا الموقع الذي تستطيع من خلاله التعديل على sources.list الى المصادر المناسبة .

```
http://www.ubuntu-nl.org/source-o-matic/
```

ولمستخدمي التوزيعات الأخرى مثل Cento او suse تستخدم هذه الأوامر لتثبيت البرامج وإزالتها.

تثبيت البرامج

```
yum install progarm
```

حذف البرامج

```
yum remove progarm
```

2. مثبت الحزم dpkg

هذا البرنامج يستخدم لتثبيت الحزم الجاهزة و المحملة على الجهاز وهي الحزم الخاصة بالدبيان debian و بعد تحميل الحزمة ذات الامتداد deb يتم التركيب بهذا الأمر :

```
sudo dpkg -i program.deb
```

وهناك مواقع متخصصة للبحث عن حزم deb منها :

```
http://www.debian.org/distrib/packages
```

ولتوزيعة ubuntu

```
http://packages.ubuntu.com/
```

و للتوزيعات التي تعتمد على حزم rpm

```
http://rpmfind.net/
http://rpm.org/
```

ولكل التوزيعات الكبيرة لها مصادر خاصة .

ويتم تثبيت الحزم rpm بهذا الأمر وأنت Root

```
rpm -i program.rpm
```

3. التركيب من المصدر

تعد هذه الطريقة هي الأصعب و الأكثر تعقيدا في احيان كثيرة و هي تركيب البرامج من الكود المصدري ،ويعرف غالبا أن البرنامج وحتى نظم التشغيل تمر بمراحل عدة وهي :

1- تحليل البرنامج
2- كتابة الكود المصدر
3- ترجمة البرنامج وتحويله الى نظام ثنائي Binary
4- استخدام البرنامج

تقوم هذه الطريقة بعملية التحويل من كود مصدري الى برنامج قابل للتشغيل و من ثم استخدامه .

سوف اشرح الطريقة الشائعة للتركيب من المصدرلان بعض البرامج لها طرق خاصة للتركيب .

بعد تحميل البرنامج و هو على شكل ملف مضغوط gz.tar في الغالب نقوم باستخراج الملفات المضغوطة.

وهذا جدول لطريقة استخراج الملفات من أغلب الامتدادات المستخدمة .

نوع الملف	أمر الاستخراج
tar.gz	tar -zxvf program.tar.gz
tgz	tar zxf program.tgz or gunzip -c program.tgz
bz2	tar jxvf program.tar.bz2
zip	unzip program.zip
rar	rar e program.rar

الآن بعد استخراج الملفات المضغوطة نقوم بالدخول إلى المجلد الناتج منه:

```
cd folder
```

folder اسم مجلد البرنامج المستخرج.

ثم نقوم بعملية الإعداد

```
./configure
```

بعد ذالك ندخل هذا الأمر

```
make
```

ثم وأنت Root تدخل هذا الأمر

```
make install
```

واذا لم يكن البرنامج يتطلب أي مكتبات خارجية سوف يتم تثبيته بشكل سليم .

هناك برامج تأتي بهذا الامتداد bin وهي جاهزة للتثبيت فقط ادخل هذا الأمر

```
./program.bin
```

تركيب برامج الويندوز في لينكس

رغم اني من المعارضين لاستخدام برامج الويندوز في لينكس لكن لا مانع من شرح تركيب برامج الويندوز في لينكس .
لتركيب أو تشغيل برامج الويندوز في لينكس هناك طريقتان :

1- من خلال برامج المحاكاة مثل wine و CrossOver .
2- عن طريق تشغيل نظام الويندوز كاملا داخل اللينكس .

تشغيل و تركيب برامج الويندوز من خلال wine :

يعد هذا البرنامج من أشهر برامج المحاكاة وهو مجاني ومفتوح المصدر ما عدا الملفات الخاصة بالويندوز، وهو يقوم بتركيب كرنل الويندوز داخل اللينكس ويطلب ملفات الويندوز أثناء تشغيل البرنامج.

ولتثبيته استخدم هذا الأمر في توزيعات debian

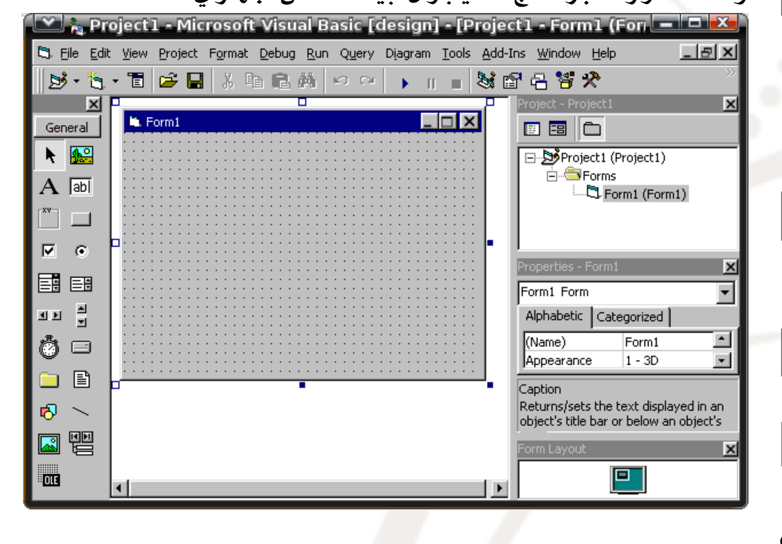
```
sudo apt-get install wine
```

وهذا موقعه

http://www.winehq.org

و تستطيع من خلاله تشغيل معظم برامج الويندوز و بعض الالعاب مثل كونتر ستريك .

وهذه صورة لبرنامج الفيجول بيسك من جهازي



وأعتقد أنه يستطيع تشغيل جميع برامج الويندوز داخل اللينكس إذا توفرت المكتبات اللازمه.

وهذا موقع فيه ملفات cox قد تحتاجونها في يوم من الايام

http://www.dlldump.com/dll-files/O_1.html

طريقة تشغيل نظام الويندوز في لينكس :

وهو باستخدام برامج الاجهزة الافتراضية مثل برنامج Virtualbox .

وتسطيع من خلاله تشغيل نظام الويندوز كاملا وحتى تشغيل توزيعات اللينكس الأخرى.

تحميل البرنامج لتوزيعة ubuntu

```
sudo apt-get install virtualbox
```

***

سطر الأوامر

يكتبه :محمد عبدالله

نكمل معكم من جديد وكما عودناكم في هذا الباب والذي نتطرق فيه الى مجموعة من أكثر الأوامر شيوعاً واستخداماً .

وتكملة لما سبق سنستعرض معكم اليوم الأوامر التالية :-

(grep – cat – head – tail – echo - sort)

1 – الأمر grep :

وهو أمر يعالج الأسطر ، حيث يقوم باستخراج النتائج منها وفق الشروط التي تضعها . وصيغته كالتالي :

```
grep options what-you-want filename
```

حيث :

what-you-want هي الشيء التي تريده أن يبحث عنه .

Filename هو الملف الذي تريده أن يبحث فيه .

Options قد تكون احدى **الخيارات التالية :**

-c حيث يعرض لك عدد النتائج التي وجدها .

-i حيث يقوم بتجاهل حالة الأحرف (كبيرة أو صغيرة) .

-n يقوم بعرض أرقام الأسطر التي تحتوي على النتائج المتطابقة .

-q لا يعرض لك نتائج (مفيد في حاله كتابة سكريبتات فقد تكون تريد اختبار نتيجة معينة ولا تريد عرض ناتجها لتقوم بإجراء معين مثلاً هل يوجد quota في ملف fstab ؟ اذا كانت النتيجة صحيحة فقم بتشغيل quota وهي المسؤولة عن تحيد مساحة للمستخدمين لا يقومون بتجاوزها)ر

-r حيث يقوم بفحص المجلدات الفرعية أيضاً بكل ملفاتها لعرض النتائج لك .

-s لا يقوم بعرض رسائل خطأ ، مثلاً في حال كان هناك ملف ليس لك صلاحية بقراءة محتواه .

-v يقوم بعكس النتائج ، أي التي لا تتطابق مع نتيجة البحث .

-V لعرض اصدارة برنامج grep .

-w للبحث عن كلمة معينة ، فالوضع الإفتراضي أن grep يعرض لك حتى الأسطر التي تكون الكلمة فيها جزة من كلمة اخرى فمثلاً عند البحث عن كلمة cat بدون استخدام هذا الخيار قد تحصل على أسطر تحتوي category ، هذا الخيار سيكون هو الحل الأمثل .

بعض الأمثلة التوضيحية :

```
grep -i jeddah frinds.txt
```

سيتكون مخرجات الأمر السابق هو كل أصدقائي الذين يسكنون في مدينة جدة ، لاحظ انني استخدم اهمال حالة الأحرف مما يعني أني قد أجد نتائج مثل Jeddah أو JEDDAH . قد يفيدني وضع الخيار n لعرض أرقام الأسطر ولكن المشكلة أنه ليس لي العديد من الأصدقاء هناك !!! لذا فستكون النتائج تعد على أصابع اليد الواحدة .

من الممكن وكما عرضنا في عدد سابق استخدام خاصية اعادة التوجية لتوجية المخرجات الى ملف آخر فمثلاً .

```
grep -i jeddah frinds.txt > frinds-from-jeddah.txt
```

مثال أخر ، لنفرض ان لدي جهاز يعمل كراوتر وعليه بروكسي squid وهذا الملف لمن لا يعرف فيه قرابة 4500 سطر ، اريد وبسرعة أن استعرض كل الخيارات التي قمت بعملها على هذا الملف وعرض النتائج ليتسنى لي معرفة كيف تسير الأمور ، سيكون الأمر التالي هو ما اريده بالظبط :

```
grep -v ^# /etc/squid/squid.conf | uniq
> /root/squid.conf.sample
```

شرح الأمر هو كالتالي :

قمنا باستخدام الخيار v والذي يعني عرض النتائج التي لا ينطبق عليها عليهاالشرط . كلنا يعرف أن الأسطر التي تبدأ بـ # يتم تجاهلها ، والعلامة ^ تعني بداية ، اذا يكون معنى المقطع الأول من الأمر هو " اعرض لي كافة الأسطر التي لا تبدأ بـ # أي الأسطر التي قمت أنا بكتابتها و هي فعلاً ما اريد " لو لم امرر مخرجات الأمر الى الأمر uniq لحصلت على نتائج تحتوي على صفحات كثيرة فارغة فعندما تحصل على 30 سطر من أصل 4500 فكم سيكون الفراغ لديك ؟؟!! إذا فالأمر uniq يقوم بحذف الأسطر المتشابهه وهي الفراغات ، ثم بعد ذلك أقوم بتمرير النتائج الى ملف باسم squid.conf.sample في مجلد الهوم الخاص بي ليتسنى لي قراءته على مهل .

ملاحظة : الأمر grep لا يقوم بتغيير محتويات الملف الذي تقوم بالبحث فيه .

2 – الأمر cat:

يقوم هذا الأمر بعرض محتويات الملفات وغالباً ما يستخدم لتمرير محتويات الى أمر ما مثل grep أو sort أو uniq .

هذا الأمر أيضاً لا يقوم بتعديل الملف وقد يكون أفضل خياراته هو n الذي يقوم بترقيم الأسطر .

أحد الأمثلة الشائعه لإستخدام هذا الأمر هو انشاء ملفات للنظام بأمر واحد فقط فمثلاً اريد انشاء ملف بإسم linux-distro.txt والكتابة فيه الى أن أكتب كلمة EOF والتي هي اختصار End Of File أي نهاية الملف :

```
cat > linux-distro.txt <<EOF
```

الآن قم بكتابة ما تريد ، ضع أسطر فارغه ومسافات وعند الانتهاء والرغبة في حفظ الملف اكتب EOF واضغط Enter لتجد الملف جاهز .

يوجد أمر آخر و هو عكس الأمر cat واسمه tac حيث يقوم بعرض الملف من الأسفل الى الأعلى .

3 -الأمر head والأمر tail .

وهي لعرض جزء من الملف فالأمر الأول يقوم بعرض أول 10 أسطر من الملف ، والأمر الأخير وكما يظهر من اسمه يقوم بعرض آخر 10 أسطر ، يمكنك تحديد عدد الأسطر عن طريق الخيار n- كما يمكنك معاينة محتوى الملف بصورة تفاعلية عن طريق الخيار f- انظر المثال التالي :

```
tail -f /var/log/message
```

سيعرض لي النظام شاشة بآخر 10 أسطر من الملف ، لاحظ أن التيرمنال لا تعود الى prompt مره اخرى ، الآن افتح طرفية اخرى وجرب اعادة تشغيل خدمة ما لترى الناتج بنفسك . وعرض آخر 10 أسطر من ملفات الـ logs أسهل حيث يمكنك تحديد سبب المشكلة عن طريق قراءة الأسطر الأخيرة ، فقد يكون الملف يحتوي على آلآف الأسطر مما يسبب صعوبة في التنقل من بداية الملف الى نهايته .

4 - الأمر echo:

يقوم هذا الأمر بطباعة جملة أو كلمة وعرضها على جهاز الخرج القياسي " الشاشة " وأكثر ما يستخدم في مجال كتابة سكريبتات الشيل ، ولكنه يستخدم أيضاً لعرض قيم متغيرات النظام مثل :

```
$LOGNAME
$PATH
$MAIL
```

كما يستخدم أيضاً لطباعة بيانات معينة واظافتها الى ملف ما فمثلاً لدي شبكة صغيرة ولست بحاجة الى اعداد DNS للتعامل من الأجهزة كأسماء بدل أرقام الآي بي ، لدي جهاز باسم backup ورقم الآي بي له 192.168.1.100 . أرغب في اضافته الى ملف hosts لدي ، اذظر المثال التالي :

```
echo “192.168.1.100            backup”
>> /etc/hosts
```

فلست بحاجة الى فتح الملف بمحرر نصوص ثم الانتقال الى نهاية الملف ثم كتابة البيانات ومن ثم حفظ الملف والخروج منه .

5 – الأمر sort :

كما يظهر من اسمه ، يقوم هذا الأمر بترتيب الملف ، فمثلاً لدي ملف يحتوي على كلمات كثيرة وكل كلمة في سطر ، ارغب في ترتيب محتويات الملف أبجدياً ، فما العمل ؟

الحل هو استخدام هذا الأمر . وهذه هي صيغته .

```
sort filename
```

وكما في الأوامر السابقة ، هذا الأمر لا يقوم بتعديل محتويات الملف ، سيكون من المفيد بعد ترتيب الملف ارسال نتائجة الى ملف جديد عن طريق > .

الأمر sort يقوم بترتيب الملف تصاعديا ، فلو أردت عكس الترتيب استخدم الخيار r .



انهاء برنامج لايستجيب

يكتبه:ياسر عمار

يمكن لمستخدمي الويندوز إنهاء برنامج عند عدم استجابتهاعتدنا عند استخدام الويندوز وتوقف برنامج ما عن الاستجابة أن نضغط الأزرار الثلاثة معا Ctrl+Alt+Delete فنختار اسم البرنامج ثم ننقر على "إنهاء العملية".

أما في لينكس فإن إننا لا نحتاج إلى ذلك ، لأن إدارة لينكس للبرامج وتنظيمه للذاكرة أفضل من الويندوز، ولكن في أحيان قليلة قد نحتاج إلى ذلك، والطريقة هنا مختلفة ولفعل ذلك نحتاج إلى الخطوات التالية:

1- معرفة اسم العملية التي نريد إغلاقها، ولعرض قائمة بأسماء العمليات في سطر الأوامر نكتب:-

```
ps -A
```

سيعرض لنا أسماء العمليات الحالية كلها في أقصى اليمين وأرقامها في أقصى اليسار كما في الشكل التالي :-

```
6319 ?        00:00:00 rssservice
6342 ?        00:00:00 korgac
6531 pts/0    00:00:00 su
6542 pts/0    00:00:00 bash
6776 ?        00:00:00 kio_uiserver
6923 ?        00:00:01 beagled-helper
7127 pts/0    00:00:24 gimp
7128 pts/0    00:00:00 script-fu
7129 ?        00:00:00 ksnapshot
7164 ?        00:00:00 kio_file
7178 ?        00:00:00 kaffeine
7179 ?        00:00:00 kio_file
```

2- إنهاء العملية باستخدام أمر (kill) متبوعا برقم العملية المراد إنهاؤها، فمثلا لإنهاء برنامج kaffeine حسب رقمه في الصورة نكتب الأمر التالي

```
kil 7178
```

ستجد أنه قد تم إنهاء البرنامج.

إذا وجدت أن البرنامج ما زال يعمل، اكتب

```
kill -9 7178
```

وإضافة 9- هي إشارة تعني إنهاء البرنامج بالقوة، وهي أشد أنواع الإنهاء (استخدمها بحذر)، والأرقام الأقل تشكل إشارات أقل قوة مثل 1 و 2.

المستندات

يتسائل بعض المستخدمين الجدد عن مكان مجلد المستندات (المستندات My Documents) الذي اعتادوا وضع ملفاتهم الشخصية فيه على ويندوز، والحقيقة أن لينكس يوفر مجلدا خاصا لكل مستخدم يحوي كل ما يخص هذا المستخدم، وهذا المجلد يحمل اسم المستخدم ويقع ضمن مجلد home فإذا كان اسم المستخدم مثلا ali فإنه مساره سيكون

```
home/ali/
```

ويرمز لهذا المجلد بالرمز (~)،

(حيث يمكن الدخول إليه من سطر الأوامر بكتابة cd ~)

وبإمكان المستخدم أن ينشأ عددا من المجلدات فيه ليرتب ملفاته الخاصة، كمجلد للمستندات وآخر للصور، ثم يضع اختصارات لها على سطح المكتب.

فحص الأقراص

يحتاج القرص الصلب إلى فحص لأقسامه، لا سيما عند حدوث مشاكل في القراءة والكتابة، ولا يخفى على من استخدم ويندوز أنه عند إيقاف تشغيل الحاسوب بشكل خطأ فإنه يتم فحص الأقراص تلقائيا أثناء تشغيله، كما يمكن ذلك يدويا من خلال خصائص القرص من جهاز الكمبيوتر، ولفحص قرص باستخدام اللينكس، استخدم أمر fsck متبوعا باسم القرص الذي تريد فحصه، وفي الغالب يكون اسم القرص /dev/sda1 للقسم الأول (وهو ما يقابل C في ويندوز)، /dev/sda5 للقسم الثاني، /dev/sda6، وهكذا .... ويمكن معرفة الأقسام الموجودة في جهازك (مع أنواعها) بكتابة الأمر

```
fdisk –l
```

ولفحص القسم الأول نكتب:

```
fsck -y /dev/sda1
```

وتضاف y– حتى لا يتوقف للسؤال عند حدوث خطأ ما، بل يقوم بالفحص تلقائيا، ويجيب عن الأسئلة بنعم.

ملاحظة 1: ينفذ هذا الأمر في وضع المستخدم الجذر، ويمكن الانتقال من المستخدم العادي إلى الجذر بكتابة أمر su.

ملاحظة 2: لا ينبغي فحص قسم موصول(mounted)، أي أنه مفتوح يمكن العمل فيه، كالذي يحتوي التوزيعة التي تستعملها، وسيعطيك تحذير، ولفحص القسم الذي يحتوي التوزيعة، نكتب:

```
init 1
```

فيدخل إلى و ضع المستخدم الواحد (Single user mode)، وعند ذلك يمكن فحص القسم بأمان.

******

تلميحات سريعة :يكتبها : الشريف أحمد حسن خلف و محمد عبدالله

تحويل صيغ برنامجNero

للتحـويل من صيغة nrg التى تخص برنامج النيرو Nero إلى صيـغة iso تحتاج إلى برنامـج nrg2iso و هو موجود بمـصادر التوزيعات المـعتمدة و كيفية إستخدامه قم بتنفيذ الأمر من سطر الأوامر فى مكان المجلد الذى يحوى الملف المراد تحويله

nrg2iso filename.nrg filename.iso

سم الملف الأول filename.nrg هو الملف المراد تحويله إلى صيغة iso و الإسم الثانى هو إسم الملف الذى سيتم إخراجه و لضم الملف الذى تمك إخراجه بصيغة iso كى يظهر كمجلد مفكوك دون فكه أو حرقه على إسطوانة قم بالدخول كمستخدم جذر و قم بعمل مجلد جديد لمكان الضم فى أى مكان تشاء

mkdir /mnt/isoimage

و أفتح سطر الأوامر فى مكان وجود الملف iso و قم بتنفيذ الأمر التالى مع تغيير إسم الملف filename.iso إلى إسم ملفك

/mount -o loop -t iso9660 filename.iso /mnt/isoimage

تشغيل امر معين في وقت معين

لتشغيل أمر معين في وقت معين فالخدمة المسؤولة عن هذا هي atd لابد من تشغيلها أولاً ، وإليك الآن بعض الأمثلة :

لتشغيل أمر معين في الساعة 2:00pm نفذ الآتي

at 2:00pm

الآن ستظهر لك علامة > لتبدأ بكتابة الأوامر ، مثلاً poweroff لإطفاء الجهاز . بعد الإنتهاء من كتابة الأوامر اضغط ctrl+d . يمكنك أيضاً تحديد الأمر بعد فترة معين مثلاً بعد10 دقائق باستخدام الصيغة التالية :

at now + number unit

حيث number هو عدد تحديده انت ، و unit هي احدى الوحدات التالية minutes, hours, days,weeks .

ولمزيد من التفاصيل راجع man at

اطفاء الجهاز بعد انتهاء التحميل

وبخصوص اطفاء الجهاز مثلاً بعد انتهاء التحميل فإليك الصيغة التالية وهي تنفيذ أمر ما بعد نجاح تنفيذ الأمر الأول

command1 && command2 && command3

وهي تعني نفذ الأمر الأول وعندما ينتهي بنجاح نفذ الأمر التالي وهكذا ، وعدد الأوامر يبدأ من 2 الـى مالانهاية .

wget -c -t 0 -T 10 http://www.example.com/filename.zip && poweroff



لكم الله يا اهل غزة,,,

فقلوبنا معكم

WWW.LINUX-AR.ORG